QoS-Aware Clouds

Fabio Panzieri, Michele Pellegrini, Elisa Turrini
Department of Computer Science
University of Bologna
Mura Anteso Zamboni 7
40127 Bologna (Italy)
panzieri@cs.unibo.it pellegri@cs.unibo.it elisa.turrini7@unibo.it

Summary

- Motivations
- QoS issues in cloud computing
  - Role of SLA
  - Earlier work
- Proposed architecture
- Experimental evaluation results
- Conclusions and future developments

Motivations

- **Cloud Computing**
  - *A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.* [I. Foster, Y. Zhao, I. Raicu, S. Lu,"Cloud Computing and Grid Computing 360-Degree Compared", in Proc. IEEE Grid Computing Environments Workshop, Austin (Tx), Nov. 2008, pp. 1-10.]
  - Software as a Service (SaaS)
    - Platform as a Service (PaaS; e.g., Microsoft Azure)
    - Infrastructure as a Service (IaaS; e.g., Amazon AWS)
- **QoS**: crucial factor for the success of cloud computing providers
  - if not delivered as expected, it may tarnish provider's reputation

QoS in Clouds

- Compliance to SLA
- SLA
  - legally binding contract stating the QoS guarantees required by cloud customer
    - typically includes max response time, throughput, error rate
    - may include non functional requirements such as timeliness, scalability, availabilitiy
      - in this work we have addressed response time, only
- QoS in clouds not sufficiently investigated as yet (t.t.b.o.o.k.)
  - growing interest in both industry and academia

Earlier work

- TAPAS
  - **T**rusted and QoS-**A**ware **P**rovision of **A**pplication **S**ervices
    - IST Project N. IST-2001-34069
      - G. Lodi, F. Panzieri, D. Rossi and E. Turrini, “SLA-Driven Clustering of QoS-Aware Application Servers”, IEEE Trans. on Soft. Eng. 33(3), pp.186-197, 2007
- RESERVOIR
  - **RE**sources and **SER**vices **V**irtualization with**O**ut Barr**I**e**R**s
    - FP7-ICT-2007-1-Objective 1.2 Project N. 215605

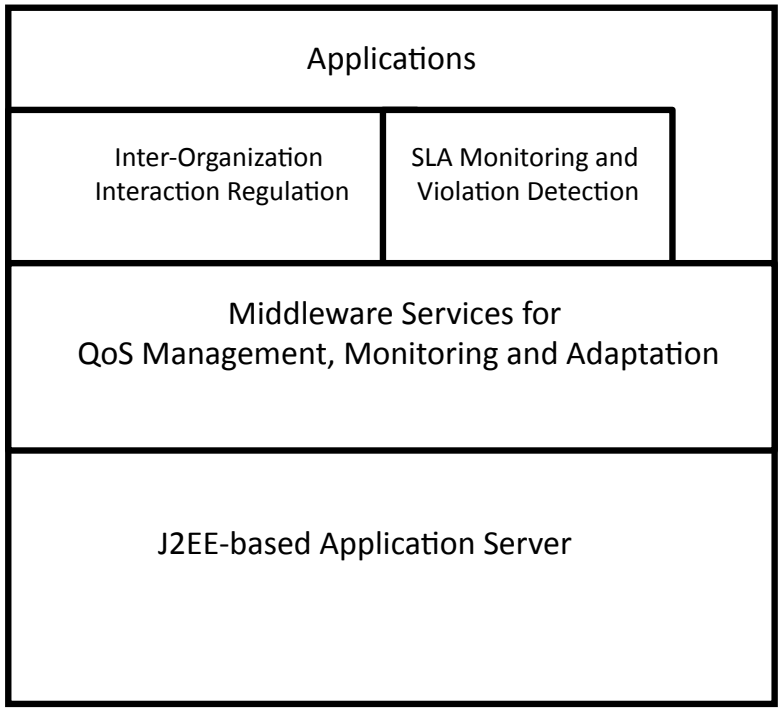
TAPAS Objectives

- Design and development of QaAS
  - QoS-awarness
    - ability to meet Quality of Service (QoS) application requirements, as specified in hosting SLA
    - hosting SLA binds hosting environment to the applications it hosts
  - Current AS technology not fully instrumented to meet those requirements
    - i.e., not designed to be QoS-aware
  - true for cloud computing technology as well
- TAPAS developed family of middleware services that make J2EE-based technology QoS-aware
  - Specifically:
    - current J2EE-based AS technologies (JBoss, WebSphere, etc…) support clustering of AS instances for scalability, load balancing, fault-tolerance purposes
    - QoS-aware AS Cluster
      - QoS-aware application hosting environment constructed out of clustered ASs

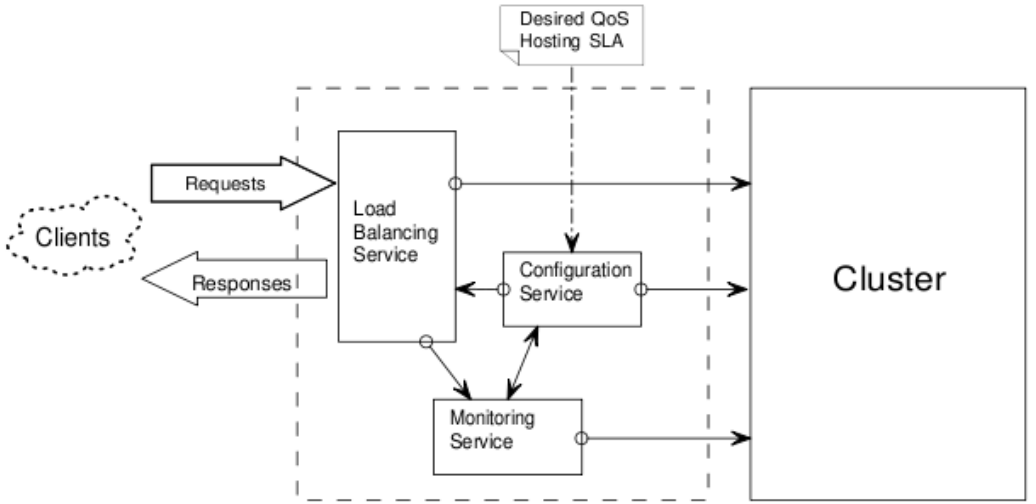
SLA enforcement and monitoring

- Carried out by two principal middleware services
  - Configuration Service (CS)
    - responsible for both configuration and (possibly) run-time re-configuration of the application hosting environment
  - Monitoring Service (MS)
    - responsible for
      - run-time monitoring of the hosting environment in order to detect possible deviations of the delivered QoS from that specified in the SLA
      - requesting application hosting environment reconfiguration, if delivered QoS deviates from SLA
- Complemented by adaptive Load Balancing Service (LBS)
- CS, MS, LBS
  - incorporated into current application server technology
  - operate both on single AS and cluster of ASs

TAPAS Architecture



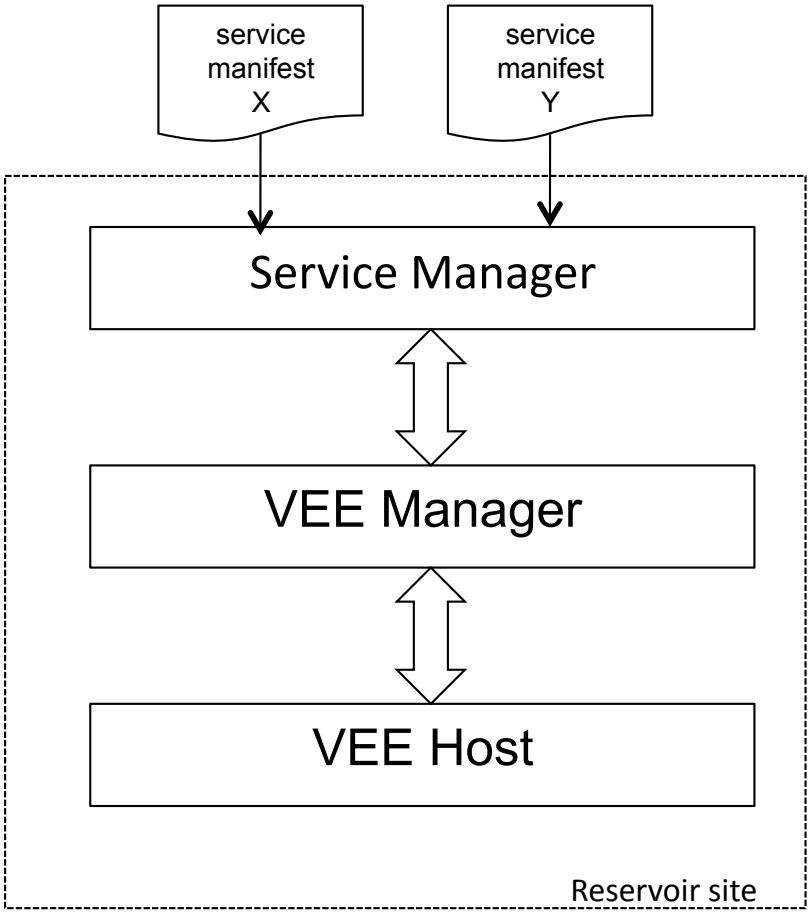
Node architecture



Middleware for QoS management

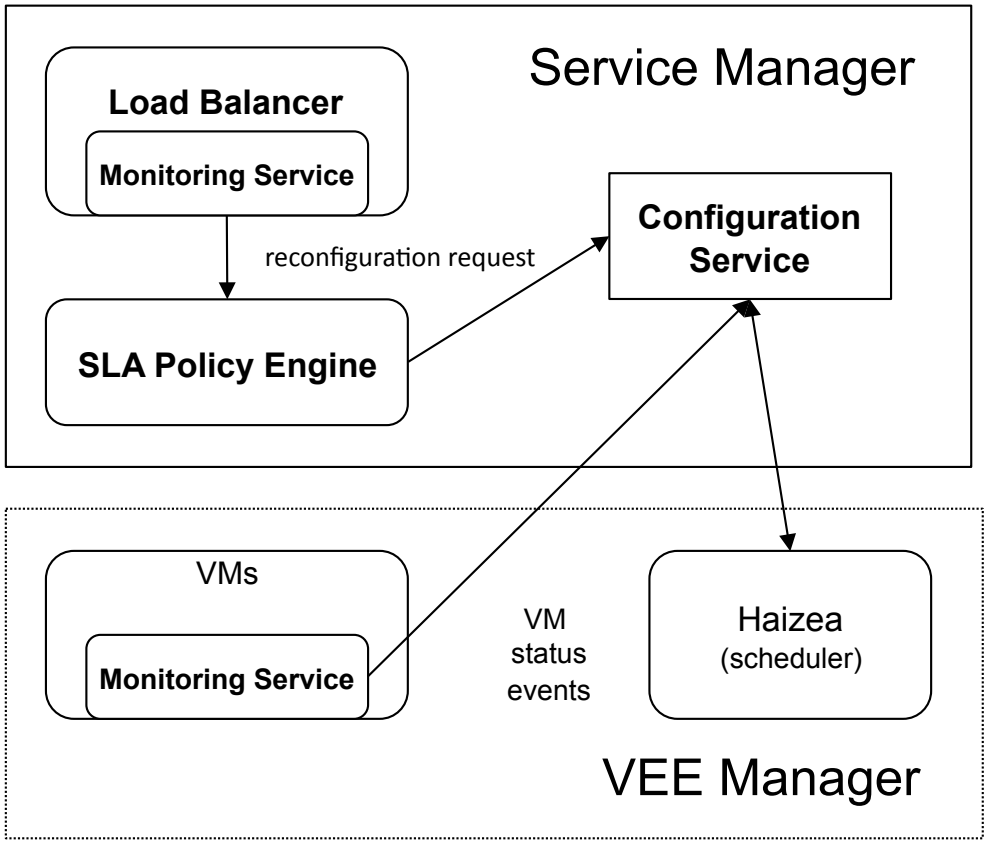
Reservoir

- Principal objectives
  - Support for cloud federations
  - Interoperability
  - Business service management
- Three levels architecture
  - Service Manager
    - application deployment based on *service manifest* (SLA)
  - VEE Manager (VEEM)
    - management and coordination of VEE Hosts
  - VEE Host (VEEH)
    - resource monitoring and control regardless of VM technology (Xen, VMware, KVM, etc.)
  - Currently, only VEEM and VEEH have been implemented



QoS-aware cloud architecture

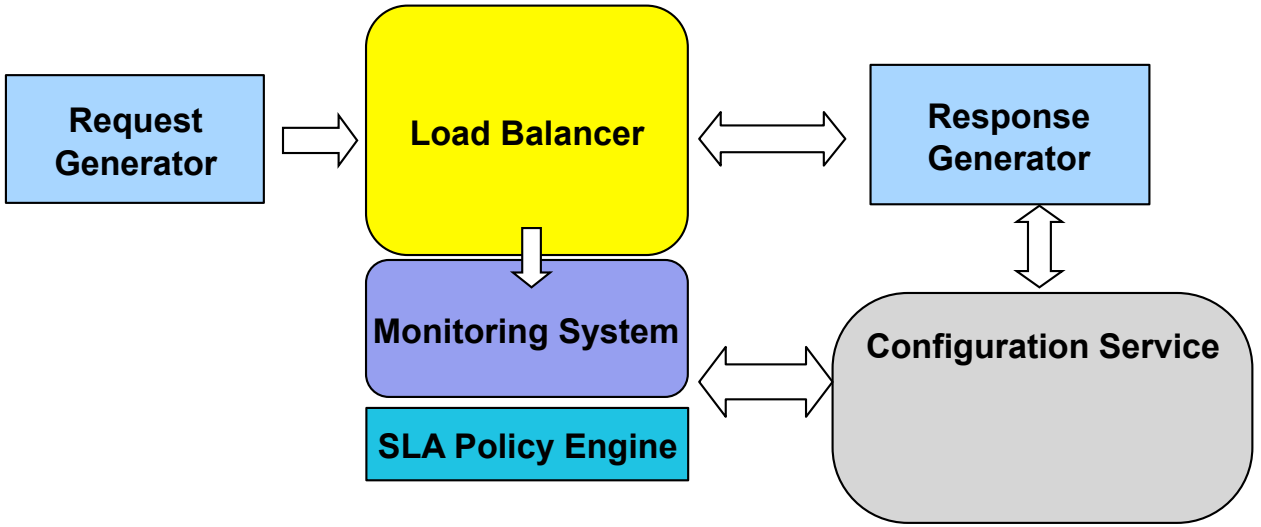
- Approach: to extend RESERVOIR Service Manager Level with TAPAS-like middleware services
- Principal components:
  - Load balancer
  - Monitoring service
  - SLA policy engine
  - Lifecycle manager



Experimental evaluation

- Scenario
  - IAAS
  - pool of available (free) VMs
  - VMs instantiated and executed on demand
  - Each VM has fixed quantity of resources (CPU, RAM, storage,…)
  - scalable services can be executed on demand
  - “pay as you go” accounting and billing model
- Scope
  - assessment of max resource allocation time
    - in order to enable development of dynamic configuration and load distribution policies that
      - optimize resource usage (no over-provision)
      - do not violate SLA
- Difficult to carry out
  - Implementation of proposed architecture
  - Unavailabiltiy of necessary IAAS infrastructure
- Evaluation carried out via simulation

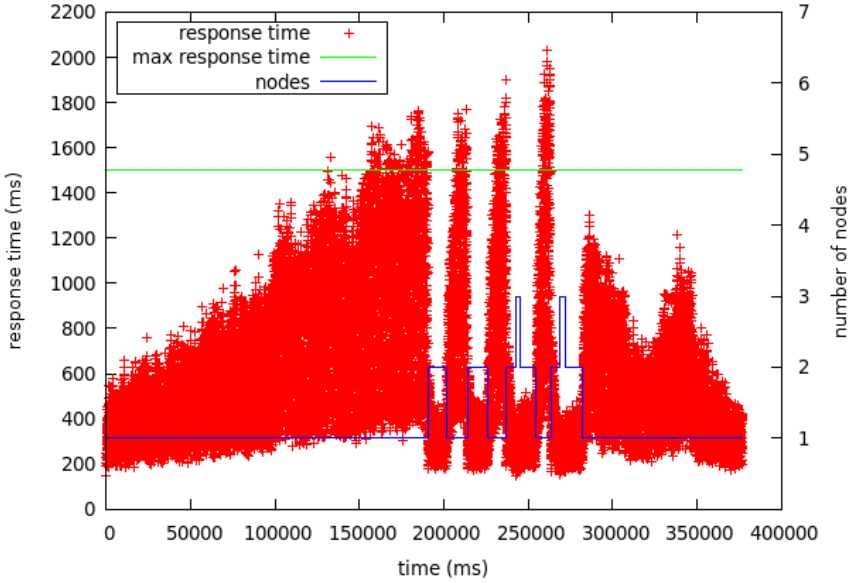
Simulation tool



- Tool implements *principal components*
  - Load balancing policies
  - QoS handling policies

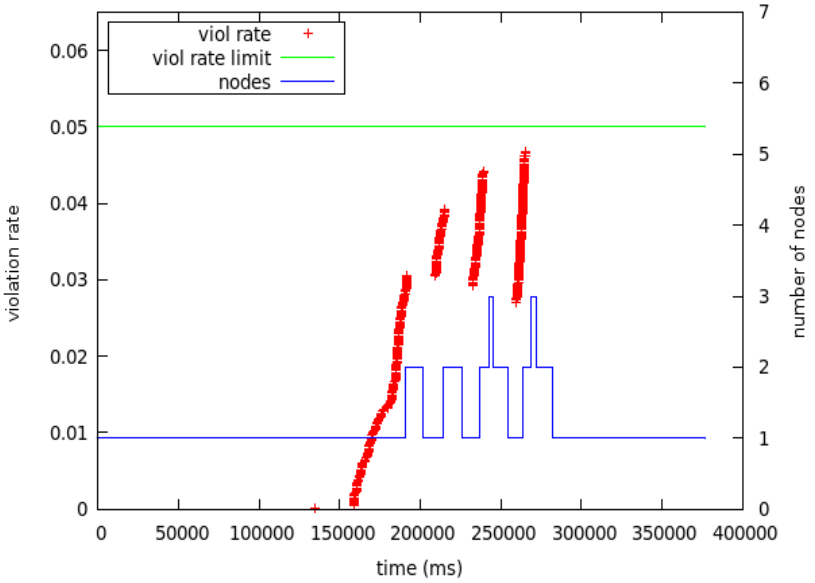
Experimental results

- **Assumptions: hosting SLA efficency = 95%, VM allocation time = 2s**
- **Other tests carried out with VM allocation time = 6s, 10s**
- **VM allocation can take up to 400s [Sotomayor B., et. al., "Overheads matters: A Model for Virtual Resource Management", Proc. VTDC 06 Workshop]**



a) Response time

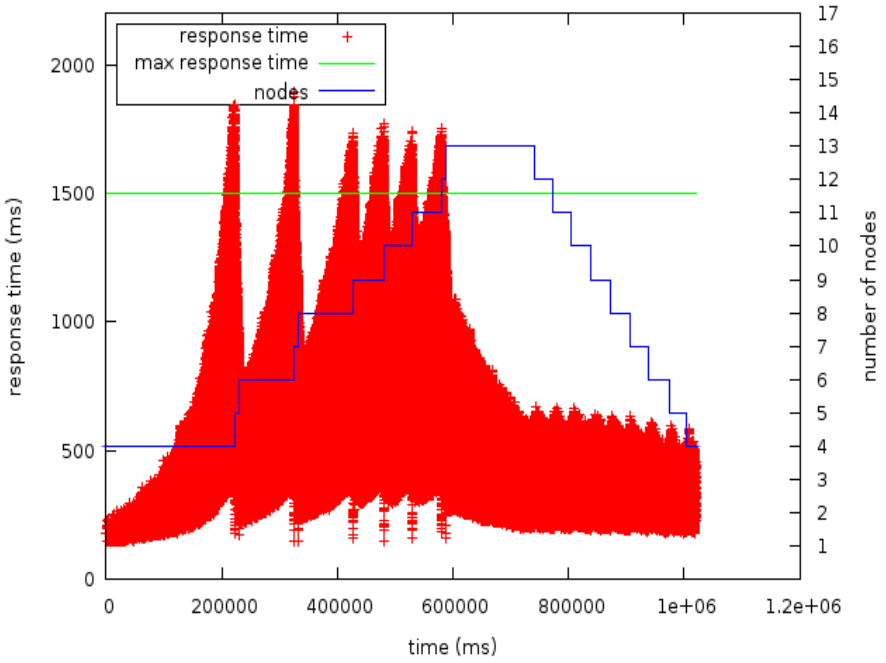
- Load up to 90 rps (SLA limit = 100rps)
- VMs allocated as load increases and released as it decreases



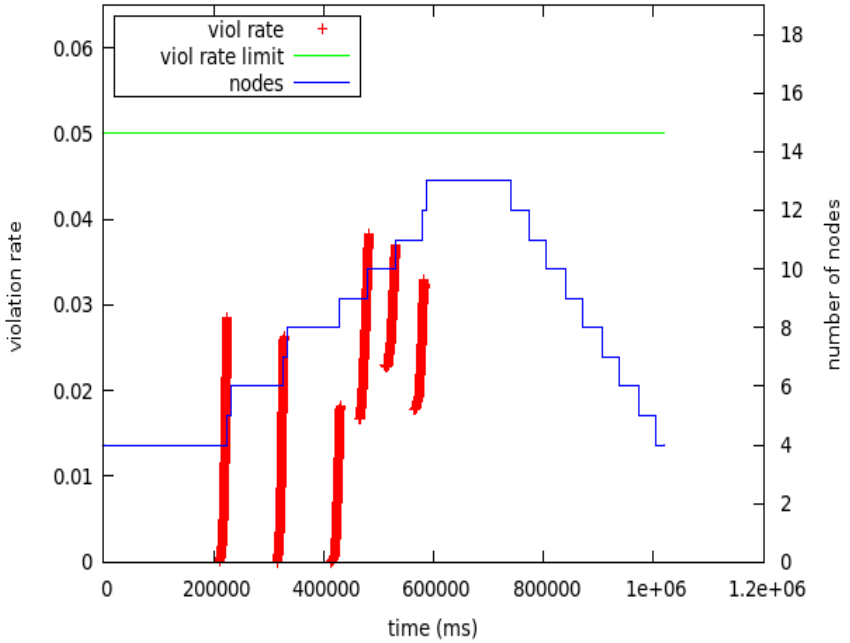
b) Violation Rate

- Peaks occur when a new VM is added

Experimental results



a) Response time



b) Violation Rate

- Peaks occur when a new VM is added

Concluding remarks

- Initial results appear to be encouraging
  - adequate design approach; however, a number of problems remain open
    - large n. of VMs may give rise to scalability problems in separate/collateral subsystems (e.g. a shared DBs may become a bottleneck)
    - VM allocation time may cause SLA violations
- Further testing required using real cloud as a test bed
  - e.g., Open Nebula (http://www.opennebula.org/), Microsoft Azure (?)
  - VM management and allocation policies that do not cause SLA violations
- Evaluation of additional QoS requirements
  - dependability requirements
- Analytical modeling
- Future developments
  - Cloud federations
  - Trust management
  - Integration of cloud computing environments and mobile devices/services